International Journal of Mechanical and Production
Engineering Research and Development (IJMPERD)
ISSN(P): 2249-6890; ISSN(E): 2249-8001
Vol. 8, Issue 3, Jun 2018, 1121-1132

TRANS STELLAR
• Journal Publications • Research Consultancy

SINGLE OBJECTIVE FOR AN INTEGER PARTIAL FLEXIBLE OPEN SHOP SCHEDULING PROBLEM USING DEVELOPED ANT COLONY OPTIMIZATION

G. UMASHANKAR[1] & D. SARAVANAN[2]

[1]*Research Scholar of Mathematics, Bharathiyar University, Coimbatore, India*

[2]*Professor of Mathematics, Karpavinagayakar College of Engineering, Madhuranthagam, Tamil Nadu, India*

ABSTRACT

As an augmentation of the classical mechanical open shop scheduling problem, the Integer partial flexible open shop scheduling problem (IPFOSP) assumes an important role in genuine production systems. In IPFOSP, an operation is permitted to be prepared on in excess of one elective machine. It has been turned out to be an emphatically NP-hard problem. Ant colony optimization (ACO) has been turned out to be a successful approach for managing IFOSP. Since, the key ACO has two essential bothers including low computational efficiency and local ideal. In defect these two brothers, a developed ant colony optimization (DACO) is proposed to propel the make span for IPFOSP. The accompanying perspectives are done on our developed ant colony optimization algorithm: select machine govern problems, instate uniform appropriated mechanism for ants, change pheromone's coordinating mechanism, select hub method, and refresh pheromone's mechanism. The genuine production instance and two plans of well –known benchmark instances are inspected and correlations with some unique approaches conform the viability of the proposed DACO. The results reveal that our proposed DACO can give better arrangement in a sensitive computational time.

KEYWORDS:- DACO, IPFOSP & Ant colony optimization (ACO)

Received: Apr 08, 2018; **Accepted:** May 17, 2018; **Published:** Jun 19, 2018; **Paper Id.:** IJMPERDJUN2018117

Original Article

INTRODUCTION

Scheduling problem assumes an essential part in numerous industrial systems [1]. In this best approach it has amazing considerable researches for late decades [2–7]. Open shop scheduling problem (OSP) is a branch of production scheduling and combinatorial optimization issues [8]. The Integer flexible open shop scheduling problem (IPFOSP) is an growth of the open shop scheduling problem (OSP) [9]. Not the same as OSP, an operation can be handled on more than one candidate machines in IPFOSP. Therefore, two sub problems facing IPFOSP are machine task and activity sequencing. Machine task is is the manner by which to dole out a machine for each activity while activity sequencing is the way to schedule all tasks on machines to optimize the given performance indicators [10]. In this way, IFPOSP is more entangled than the established OSP and it has been ended up being a firmly NP-hard in 1993 [11]. The IPFOSP was first studied by Brucker and Schliew housed a polynomial approach to deal with two opens IPFOSP [12].

In recent years, a large number of heuristics or meta-heuristics have been active to accord with IPFOSP, specifically through tabu search (TS) [13], simulated annealing (SA) [14], genetic algorithm (GA) [15, 16], particle swarm optimization (PSO) [17, 18], ant colony optimization (ACO) [19], artificial bee colony(ABC)[20], and

hybrid approaches based on different heuristics and meta-heuristics. Among these meta-heuristics, ACO has been accepted to be an able access for dealing with OSPs [21–23]. Be that as it may, this access still has few confinements in practice:(1)a lot of computational time will be spent on accepting the ideal solution. (2) The search usually falls into local ideal solution. In this manner, in order to beat these constraints, various advanced ACO algorithms or hybrid ACO algorithms have been developed.

A knowledge-based ACO approach for unraveling fractional flexible open shop scheduling problems is proposed in [19]. A developed ACO was proposed to manage with dynamic hybrid flow shop scheduling in [24]. A modified ACO called two-pheromone ant colony optimization was proposed for unraveling Integer flexible open shop scheduling problem with due window in[25]. Ant colony optimization mingled with tabu search was used to deal with OSP in [26]. Two-generation Pareto ant colony algorithm was proposed by Zhao et al. for comprehending multi objective open shop scheduling problem [27]. A two-stage ant colony optimization was exhibited to minimize the make span in [28]. Leung et al. [29] proposed an agent- based ant colony optimization for tackling integrated process planning and scheduling problem. Admirable endeavors have been done for accomplishing open shop scheduling problems or Integer flexible open shop scheduling problems by developing ACO algorithms; regardless, these developed ACO algorithms are master by evolving pheromone update mechanism. In spite of the best approach the search speed and solution efficiency, excessively strengthening the pheromone criticism of the most ideal way may effectively promote to premature convergence. Therefore, in order to tackle these problems that already been in the fundamentals of ACO or developed ACO algorithms mentioned above, we propose an developed ant colony optimization (DACO) to tackle the IFOSP in this paper and the results are found to be closer or equal to the global optimum. The remainder of this paper is organized as follows. Section 2 describes the model formulation for IFOSP. The proposed DACO is introduced in Section3. Experimental test, comparison, and discussion are reported in Section 4. The conclusions and future work are given in Section 5.

Problem Formulation of IPFOSP

The n×m IPFOSP can be portrayed as follows [30]: there are n open shop and m machines. Each open shopu includesn_u operations {O_{u1}, O_{u2}, O_{u3},..., $O_u n_u$}. Each operationo_{uk}can be handled by just a single machine from the competitor machine setA_{uk}. The assumptions for IFOSP are as follows:

- Each machine can be utilized at time zero.
- Each open shop can be changed at time zero.
- Each machine can transform just a single operation at a time.
- Once an operation begins on a machine, it can’t be interrupted.
- The succession of operations for all open shops are pre specified.
- Neither due dates nor discharge times are indicated.
- The transportation times among machines are not taken in consideration.
- All machines are not generally same.

The goal is to minimize the make span and the scientific model of the FJSP is shown as follows [31]:

$$minf(x) = C_M = \max_{1\le u\le n}\{cini\}, \tag{1}$$

$$\text{s.t} \left[\left(c_{hg} - c_{uk} - t_{hgz}\right).x_{hgz} \ge 0v\left(c_{uk} - c_{hg} - t_{ukz}\right).x_{ukz} \ge 0\right], \forall u, z, g, h \tag{2}$$

$$cini - c_{u(k-1)} \ge t_{ukz}.x_{ukz}, k = 2, \dots, nu, \forall u, z, \tag{3}$$

$$\sum_{x_{ukz}\in A_{uk}} x_{ukz} = 1, \forall u, z, k, \tag{4}$$

$$x_{ukz} \in \{0,1\}, \forall u, z, k, \tag{5}$$

$$c_{uk} \ge 0, \forall u, k, \tag{6}$$

Where u, h is the open shop index, u, h = 1,2,...,n. z is the machine index, z=1,2,...,m. k, is the operation index, k,=1,2,...,n_u. n represents total open shops. m represents total machines.n_u represents total operations of open shop$u$. t_{ukz}represents processing time of k_{th}operation of open shopu on machine z. $c_{uk}$is the completed time ofo_u. x_{ukz}=1, if machine zis selected for ouk; otherwise,x_{ukz} =0. Equations (2) and (3) are the operation succession imperative. Equation (4) demonstrates that each operation can only be prepared on one machine from machine set at once. Equations (5) and (6) are choice variables which are 0-1 binary variable and non-negative, individually.

Developed Ant Colony Optimization for IPFOSP

Standards of Ant Colony Optimization

As social insects, ants live in settlements and their conduct is controlled by the objective of province survival rather than being fixated on the survival of individuals. The fundamental idea of the ACO is propelled by the conduct of genuine ants searching for food. The genuine ants can pass on with each other about nourishment sources through pheromone. The moment when those genuine ants move along, they discharge pheromone on the way they have passed. Other ants are pulled to tail them by watching the pheromone trail.

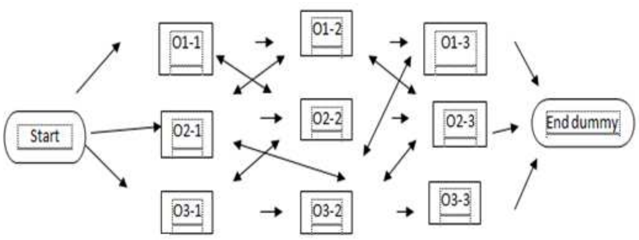

Figure 1: Ants' Moving Disjunctive Graph

Therefore, the best procedure is improved furthermore in this way attracts more ants [32, 33]. Compared with some other heuristics, the ACO is depicted eventually by distributed computation and positive feedback [34]. Though the operations are looked comparatively as ants, it can be seen successfully that there exist lots of similarities between an ant colony's for aging methodology and IFOSP. Operations must hunt for proper machine to process them. Like ants, they need to look for the shortest path. Ants' nest and foods are similar with start and end dummy operation, individually. If we gander at an operation as an ant's way for scrounging food, at that point during the perspective from claiming association of any two operations can be looked at as an elective path, furthermore different transforming period of all the operations on the machine barely like different length of paths. There are three open shops furthermore each open

shop has three operations. O1-1, O1-2, O1-3 represent three operations from claiming open shop 1. O2-1, O2-2, and O2-3 represent three operations from claiming open shop 2. O3-1, O3-2, and O3-3 represent three operations from claiming open shop 3. Each open shop must comply with the process sequence imperatives. According to the constraints of process sequence and machine occupancy, ants travel total nine operations of the three open shops furthermore scan for the operation order on each machine and then gain the ideal or near-optimal solution for the Integer flexible open shop scheduling problem. Therefore it is feasible to utilize ACO to tackle those Fractional flexible open shop scheduling problems. The key steps of the basic

ACO are the calculations of transition probability, visibility, and pheromone amount. The nodes or ambition point are selected by ants according to pheromone amount and visibility [35]. At time t, the probability for ant k allotting the path from point u to z is calculated as follows:

$$p_{uz}^{k}(t) = \begin{cases} \frac{\tau_{uz}^{\alpha}(t)\eta_{uz}^{\beta}(t)}{\sum_{s\epsilon allowed_k}\tau_{uz}^{\alpha}(t)\eta_{us}^{\beta}(t)}, z\,\epsilon allowed_k \\ 0 \\ otherwise, \end{cases} \tag{7}$$

The place the $allowed_k$ is selectable machines gathered that ant k can allot and α and β are the pheromone and expectation factor, respectively.$\eta_{uz}(t)$is heuristic factor furthermore it is computed as $\eta_{uz}(t) = 1/f(x)$. The more pheromone sum on the way is and the greater visibility is and the greater probability for allotting this way is.

As time goes on, the pheromone of the path evaporates gradually. An ant will adopt the pheromone amount on the passed edges by applying the local updating rule, as follows:

$$\tau_{uz}(t) = (1-\rho).\tau_{uz}(t) + \rho\tau_{0,} \tag{8}$$

Where ρ is the evaporation coefficient of pheromone and 0<ρ< 1. $\tau_0$is the initial amount of pheromone on each path. Once all ants have arrived at their place, the amount of pheromone on the edge is adapted again by applying the global updating rule, as follows:

$$\tau_{uz}(t) = (1-\rho).\tau_{uz}(t) + \rho\Delta\tau_{uz}^{k}(t), \tag{9}$$

Where$\tau_{uz}^{k}(t)$is the abundance of pheromone on the path(u,) laid by ant, and it can be authentic as follows:

$$\Delta\tau_{uz}^{k}(t) = \begin{cases} \frac{Q}{f(x)^k}\ if\ ant\ k\ travel\ through\ are\ (u,z) \\ 0\ otherwise, \end{cases} \tag{10}$$

In (10), Q is a constant and it denotes the backbone of pheromone; $f(x)^k$is evaluation amount of the kth ant after finishing the search task. Concerning illustration above, the fundamental ACO even has some weaknesses in practice. For example, the search generally gets trapped in local ideal result. Meanwhile it needs a lot of computational time eto access the ideal solution. Therefore, we propose a developed ant colony optimization to access the IFOSP in this paper.

Developed Aspects on Ant Colony Optimization Algorithm

The accompanying upgrades are done for machine selection: the machine which has the shortest processing time for completing the zth operation of open shopu is chosen by using 60% probability; the machine which has the shortest time for processing the zth operation of open sopu is chosen by utilizing 30% probability; the machine is randomly chosen by utilizing 10% probability, and the random choosing can be achieved by utilizing roulette selection method.

The range for selecting machine can be expanded by utilizing these three kinds of selection method. Take Figure 2, for instance, $T1<T2<T3$; therefore $M1$ is subsequently chosen with a 60% probability, $M2$ is subsequently chosen with a 30% probability, and one machine among $M1$,2 and $M3$ is subsequently chosen with a 10% probability.

The primary steps of the fundamental ACO are the introduction position of ants, the calculation of transition probability, visibility, and pheromone value, as mentioned in Section 3.1. When some other control parameters stay unchangeable, the starting position of the ants has a greater impact on the ACO. The ants ought further to be distributed uniformly on the set which contains the first operations of all open shops, and the probability for searching for the global perfect result will become much greater. If an extensive number of ants search for nourishment from the same starting point, the result diversity may be lost. Therefore, an introduction mechanism is utilized to distribute uniformly the ants' starting positions. In the starting search stage, some ways are passed by ants, and some other ways are not passed. If an ant searches for way stated to the pheromone's guiding mechanism, it is easy to decrease the probability for choosing the way that has not been passed yet, furthermore, accordingly the risk for ants to find the global ideal result will be decreased. So when the pheromone exceeds a certain value, the ants are permitted to discover the ideal way as stated to pheromone's guiding mechanism, as follows:

$$\tau_{uz}(t) \geq 1.1 * \tau_0, \tag{11}$$

The place$\tau 0$ is the starting value of pheromone on each way and τuz(t) is pheromone value between hubu and z (pathu →z) at time$t$. So as to extend the search scope of the ants and to enhance the search space of the fundamental ACO, introducing pheromone necessities to be done when the pheromone value on a way is more than 90% of the total pheromone value on all ways (shown in (12)) although the ACO has tumbled under local ideal in this situation.

$$\tau_{uz}(t) \geq 0.9 * \tau_{sum}(t), \tag{12}$$

The place $\tau_{sum}(t)$is the aggregate pheromone value on all ways. However, the following accessible way chosen by employing the transition probability does not always obtain the ideal direction for the fundamental ACO, and the pheromone deviated from the ideal solution has the potential to be enhanced, which will undoubtedly prompt to the local ideal solution. After the transition probability for every candidate is obtained, if the roulette selection method is adopted, not just the way with a expansive transition probability is likely to be chosen, as well as the way with a little transition probability has the chance to be chosen; furthermore the search space and solution quality can be extended and improved, individually. Furthermore, a new hub selection method, joining former knowledge, probability search, and random search, is proposed in this paper. When the search is trapped in local ideal solution, the result space can be further searched by changing the pheromone also expanding the random selection probabilities.

$$P = \begin{cases} p = arg\ \max_{z\epsilon allowed_k}\{[\tau_{uz}(t)]^{\alpha}.[\eta_{uz}(t)]^{\beta}.\xi_{uz}(t)\}\ 0 \leq q \leq q_0 \\ p_2 = p_{uz}^k(t) = \begin{cases} \frac{\eta_{uz}^{\alpha}(t)\eta_{uz}^{\beta}(t)\xi_{uz}(t)}{\sum_{s\epsilon allowed_k}\tau_{is}^{\alpha}(t)\eta_{us}^{\beta}(t)\xi_{uz}(t)}, j\epsilon\ allowed_k \\ 0\ otherwie \end{cases} q_0 < q \leq q_1 \\ p_3 = random\ search\ q_1 < q \leq 1 \end{cases} \tag{13}$$

The place$q$ is a random value furthermore $0 \leq q < 1$; $q0$ is the degree of former knowledge; $q1$ is the lower bound level of random search; $\xi_{uz}(t)$represents the appearing number of arc(u, z) throughout past iterations for seeking good result. The more the arc (u, z) appears, the more the role is assumed for seeking good result by utilizing positive feedback.

A best balance relation between "utilizing the past information to speed up the convergence" and "exploring new ways" can be determined by joining these three selection methods. Toward this way, the search space can be extended and the global best result can be obtained with a greater probability.

In addition, an idea of the invalid search number is defined, which intends the difference between the current number from claiming iterations$N1$ and the recent number from claiming iterations $N2$ for moving forward the result, as shown in (14). When the number of the invalid search extends a specified value $N0$, the algorithm is viewed as a local optimum, as shown in (15).

$$N = N_1 - N_2 \tag{14}$$

$$N \geq N_0. \tag{15}$$

Mean while, the maximum or minimum pheromone trails may prompt to premature merging for seeking result.

Therefore, the maximal pheromone trail τmax and the minimal pheromone trail τmin are provided in our DACO so as to make constantly on pheromone trails $\tau_{uz}(t)$fulfilτmin ≤ $\tau_{uz}(t)$≤ τmax. This idea is propelled by the Max–Min ant system [36]. When the number of the invalid searches exceeds a specified value $N0$, the pheromone on the path is forcedly destroyed in order to avoid falling into the local optimum. In this paper , the value is reduced to sixty percent to the original pheromone$\tau_{ij}(t)$, as follows:

$$\tau_{uz}(t+n) = \begin{cases} 60\% * \tau_{uz}(t) \ if \ N \geq N_0 \\ \tau_{uz}(t) = (1-\rho).\tau_{uz}(t) + \rho\Delta\tau_{uz}^{k}(t) \ otherwise \end{cases} \tag{16}$$

3.3. Steps of the DACO. The particular implementation steps of the suggested DACO for solving IFOSP are shown as follows.

Step1. Initialize parameters α,,,,0,q1, andτ0 and tabu list.

Step 2. Initialize the ants' starting positions by provisions of uniform distribution mechanism.

Step 3. Select machine by provision of our proposed machine selection strategy.

Step 4. Establish three sets: particular set S_1 holds operations that have been already visited by ants. One set S_2 holds operations for abutting candidates. An alternate set S_3 holds operations that are waiting to be added to the candidate set. And include the first operation of each open shop to the set of next candidate operation waiting to be chosen.

Step 5. Judge if the pheromone $\tau_{uz}(t)$on the way is larger or equal to1.1∗τ0 or not. Assuming that not, that point select next operation in a random manner. Provided that after select the next way according to (13). Following the transition probability for each candidate is obtained, those roulette selection method is decided to select the next way or hub.

Step 6. Include the just decided operation to setS_1 furthermore remove the barely decided operation from set S_1. Meanwhile, include the following operation to setS_2 and remove the following operation from setS_3.

Step 7. Judge if there are subsequent operations or not. If there are subsequent operations, then move on to Step 5. Or else move on to Step 8.

Step 8. Calculate the great result of this iteration and save.

Step 9. Find the best ant and upgrade its global pheromone.

Step 10. Calculate the pheromone value on each way and limit the pheromone value on each way within the range of two valuesτmax andτmin. Meanwhile, judge if$\tau_{uz}(t)$is notlessthan0.9$*\tau_{sum}$ (t). Provided, instatethe pheromone value of that way.

Step 11. Judge if$N$ is more than $N0$ or not. If it is fulfilled, then forcedly destroy the pheromone value of that way stated (16).

Step 12. End the algorithm either when the ideal or near- ideal result is found or when a maximal iteration is fulfilled and then output the global ideal or near-ideal result. Or else move on to Step 2.

Experimental Results

To test the execution of the proposed DACO, two gatherings of simulation experiments are executed. One gathering of experiments originates from a genuine production instance, and the other gathering of experiments originates from benchmark issues.

Test on the Actual Production Instance

Table 1 shows the process information from an actual production instance. The value in Table 1 represents processing time on each machine and "—" means the operation cannot be processed on that machine. The parameters and their values, which are used for running our DACO, are shown as follows: number of ants m=39, weight of pheromone trailα=1, weight of heuristic informationβ=2, pheromone evaporation parameter ρ= 0.1, the initial value of pheromone valueτ0 =0.1, constant for pheromone updating$Q$=120, premature constant$N$0 =20, τmax = 10, τmin = 0.01, the value of prior knowledge q0 = 0.3, the lower bound value of random search q1 = 0.8, and number of iterations$G$=100. The Gantt chart obtained by our DACO, the basic ACO, and Max–Min Ant System (MMAS). In addition, the merging curves of the best make span by using our DACO, the basic ACO, and MMAS. With such an instance, t can be seen that our DACO obtains the ideal make span (25) with 8 iterations. The essential ACO only obtains the ideal make span (28) and it needs 28 iterations. In spite of those MMAS might entry those ideal make span (25), however, it needs as abounding as 61 iterations. Therefore, our proposed DACO is really effective for tackling IFOSP. In order to prove the solving solution quality and solving ability four DACO, Table2 shows the experimental results of the proposed DACO in comparison to the basic ACO and MMAS by 10 absolute experiments. The best solution, the average solution, the worst solution, and the average computing time of our proposed DACO are the best among the compared methods. The results demonstrate that the proposed DACO is effective and efficient in solving IFOSP.

Test on Kacem Benchmark Instances

FOSP was classified into two categories by Kacem et al. [37]: (1) Integer FOSP (P- FOSP)which means that each operation can be processed by a subset of machines and (2) Total FOSP (T-FOSP) which means that each operation can be processed by all machines.

Firstly, one 8×8 P-FOSP and one 10×10 T-FOSP from bench mark instance sere connected to assess those execution of our proposed DACO. The Gantt graph and the merging curve for 8 × 8 benchmark P-FOSP obtained by our DACO. The Gantt chart graph and themergingcurvefor10×10 bench mark T-FOSP obtained by our DACO. With such an instance, our DACO can success fully obtains the best make span with very little iteration. In addition, we analyze the

performances of our proposed DACO with few strategies including AL + CGA [37].

Table 1: Process Information

Open Shop	Operation	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
OS1	O_{1-1}	2	–	4	–	10	3	–	8	–	5
	O_{1-2}	5	7	–	8	–	9	–	–	–	11
	O_{1-3}	–	6	5	–	10	9	–	–	4	–
	O_{1-4}	–	–	–	7	–	5	–	8	–	–
OS2	O_{2-1}	10	–	4	–	10	–	–	8	–	–
	O_{2-2}	9	–	8	–	–	7	–	6	–	–
	O_{2-3}	–	6	–	5	–	8	7	–	–	4
OS3	O_{3-1}	–	–	9	–	6	–	4	–	–	5
	O_{3-2}	–	–	6	–	8	–	–	7	–	–
	O_{3-3}	–	9	–	–	–	5	–	6	4	–
	O_{3-4}	–	–	4	–	3	6	–	–	–	–
OS4	O_{4-1}	–	8	–	5	–	–	2	4	–	–
	O_{4-2}	–	–	6	–	–	–	8	–	5	–
	O_{4-3}	5	–	–	8	–	–	4	–	–	9
	O_{4-4}	–	3	–	–	11	6	–	8	–	10
	O_{4-5}	–	–	2	–	5	8	–	–	8	–
OS5	O_{5-1}	6	–	–	4	–	3	–	–	8	–
	O_{5-2}	3	–	–	7	–	6	–	10	–	8/9
	O_{5-3}	–	–	9	–	–	–	4	–	–	11
	O_{5-4}	–	7	–	–	6	–	–	7	–	9
OS6	O_{6-1}	–	–	–	–	10	8	–	5	–	–
	O_{6-2}	–	–	–	3	–	–	2	7	–	3
	O_{6-3}	–	3	–	–	10	–	–	–	6	–
	O_{6-4}	10	–	7	6	–	10	–	–	9	8
	O_{6-5}	–	–		–	8	9	–	10	–	–
OS7	O_{7-1}	9	–	5	–	7	9	–	–	–	8
	O_{7-2}	2	–	6	–	8	-	3	7	–	–
	O_{7-3}	–	–	–	5	6	2	–	9	3	–
	O_{7-4}	9	–	15	5	3	–	–	6	–	5
OS8	O_{8-1}	–	3	–	6	–	8	5	–	–	–
	O_{8-2}	4	–	6	3	–	–	–	8	10	–
	O_{8-3}	–	5	8	–	–	6	–	–	5	–
	O_{8-4}	2	–	6	–	–	–	6	–	3	8
	O_{8-5}	2	–	–	5	–	3	–	5	9	8

Table 2: Comparison of the Algorithm Performances for the Actual Production Instance

Algorithm	The Best Solution	The Worst Solution	The Average Solution	The Average Computing Time (S)
The basic ACO	28	36	32.4	9.327
MMAS	25	28	26.8	6.458
AACO	23	24	23.5	3.925

PSO + TS [17], PVNS [38], KBACO [19], and TSPCB [13]. Table 3 shows the best results of all these several methods, where $C*$ is the best value found so far. From Table 3, it can be concluded that the proposed DACO is not worse than other algorithms, even better than several improved algorithms. Meanwhile, the proposed DACO can Almost obtain the best values for the five benchmark problems. For every independent test. In addition, the running time(RT) of the proposed DACO is very short; for instance, it can find thebestsolutionforthe10×10T-FOSP in the second option within 5s.

Table 4 illustrates the comparison of the best value and the average value between DACO and those from literatures (BEDA [39]; PBABC [40]; EA [41]; and EQEA [9]) on the Kacem benchmark instances [37]. For each instance, all algorithms are run for 10 times independently. From Table 4, it can be seen effectively that the best quality can be obtained by these algorithms with a 100% success rate. Therefore, our DACO is very effective and strong.

Test On The BR Data Instances. Next We Do Tests On The Ten Brdata Instances [11]. Table 5 Indicates The Correlation Effect

Table 3: Results of the Five Kacem Benchmark Instances

Problem	N × M	C*	AL+ CGA[37]	PSO+ TS[17]	PVNS [38]	KBACO [19]	TSPCB [13]	DACO Best RT
CASE 1	4×5	11	16	_	_	11	11	10 0.41
CASE 2	8×8	14	15	15	14	14	14	13 3.21
CASE 3	10×7	11	15	_	_	11	11	12 3.06
CASE 4	10×10	7	7	7	7	7	7	7 4.10
CASE 5	15×10	11	23	12	12	11	11	9 4.54

Table 4: Comparison Result of the Best Value and the Average Value

Problem	$n\times m$	$C*$	BEDA [39] Best Avg	PBABC [40] Best Avg	Chaing & Lin [41] Best Avg	EQEA [9] Best Avg	DACO Best Avg
CASE 1	4×5	11	11 11	11 11	11 11	11 11	10 10
CASE 2	8×8	14	14 14	14 14	14 14	14 14	13 13
CASE 3	10×7	11	11 11	11 11	11 11	11 11	12 12
CASE 4	10×10	7	7 7	7 7	7 7	7 7	7 7
CASE 5	15×10	11	11 11	11 11	11 11	11 11	9 9

Table 5: Results of the Ten Kacem Benchmark Instances

	LEGA[42]	GA[43]	PVNS[38]	KBACO[19]	TSPCB[13]	DACO
MK01	40	40	40	39	40	38
MK02	29	26	26	29	26	22
MK03	_	204	204	204	204	204
MK04	67	60	60	65	62	58
MK05	176	173	173	173	172	169
MK06	67	63	60	67	65	59
MK07	147	139	141	144	140	139
MK08	523	523	523	523	523	523
MK09	320	311	307	311	310	300
MK10	229	212	208	229	214	202

By using our proposed DACO and LEGA of Ho et al. [42], GA of Pezzella et al. [43], PVNS [38], KBACO [19], and TSPCB [13]. As for the best make spans, from Table 5, it can be seen that with the best results obtained our DACO is equal or smaller than that of other algorithms for dealing with almost all ten BR data instances. Our DACO outperforms GA [40] in three out of the ten BR data instances, outperforms KBACO [19] in six out of the ten BR data instances, outperforms LEGA [39] in seven out of the ten BR data instances, outperforms TSPCB [13] in four out of the ten BR data instances, and is almost good as PVNS [38] for the ten BR data instances. Therefore, it is concluded that our DACO has more powerful optimizing ability in dealing with partial flexible open shop scheduling problem.

CONCLUSIONS

In this paper, an efficient DACO is proposed for IFOSP in order to minimize make span. Experimental results on an actual production instance and two sets of well-known benchmark IFOSP Instances indicate that our proposed DACO is competitive to other algorithms. The results demonstrate that the proposed DACO is effective and efficient in dealing with IFOSP. Our future worth of effort needs to be done from the following aspects: (1) multi objective Integer flexible open shop scheduling problem especially related to the fuzzy due date and energy consumption need to be considered.(2)Dynamic scheduling for IFOSP may be in turn research direction, in view to real manufacturing systems, erratic events, for example, machine breakdown and new open shop arrivals and so on, often happen.

REFERENCES

1. *V. Kaplano˘glu, "An object-oriented approach for multi- objective flexible job-shop scheduling problem," Expert Systems with Applications, vol. 45, pp. 71–84, 2016.*
2. *M. Gen and L. Lin, "Multi objective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey," Journal of Intelligent Manufacturing, vol. 25, no. 5, pp. 849–866, 2014.*
3. *J. Li, Q. Pan, and S. Xie, "An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems," Applied Mathematics and Computation, vol. 218, no. 18, pp. 9353–9371, 2012.*
4. *J.-J. Wang and Y.-J. Liu, "Single-machine bi criterion group scheduling with deteriorating setup times and job processing times," Applied Mathematics and Computation, vol. 242, pp. 309–314, 2014.*
5. *L. Wang, S. Wang, and M. Liu, "A Pareto-based estimation of distribution algorithm for the multi-objective flexible job- shop scheduling problem," International Journal of Production Research, vol. 51, no. 12, pp. 3574–3592, 2013.*
6. *L. Wang, G. Zhou, Y. Xu, and M. Liu, "An enhanced Pareto- based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling," The International Journal of Advanced Manufacturing Technology, vol. 60, no. 9–12, pp. 1111–1123, 2012.*
7. *L. J. Zeballos, "A constraint programming approach to tool allocation and production scheduling flexible manufacturing systems, "Robotics and Computer-Integrated Manufacturing, vol. 26, no. 6, pp. 725–743, 2010.*
8. *I. Driss, K. N. Mouss, and A. Laggoun, "A new genetic algorithm for flexible job-shop scheduling problems, "Journal of Mechanical Science and Technology, vol. 29, no. 3, pp. 1273–1281, 2015.*
9. *X. L. Wu and S. M. Wu, "An elitist quantum-inspired evolutionary algorithm forthe flexible job shop scheduling problem," Journal of Intelligent Manufacturing, vol. 26, no. 2, pp. 1–7, 2015.*
10. *K. Z. Gao, P. N. Suganthan, T. J. Chua, C. S. Chong, T. X. Cai, and Q. K. Pan, "A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion," Expert Systems with Applications, vol. 42, no. 21, pp. 7652–7663, 2015.*
11. *P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search," Annals of Operations Research, vol.41, no.3, pp. 157–183, 1993.*
12. *P. Brucker and R. Schlie, "Job-shop scheduling with multi- purpose machines, "Computing, vol.45, no.4, pp. 369–375, 1990.*
13. *J.-Q. Li, Q.-K. Pan, P. N. Suganthan, and T. J. Chua, "A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem," International Journal of Advanced Manufacturing Technology, vol. 52, no. 5– 8, pp. 683–697, 2011.*

14. A. Jamili, M. A. Shafia, and R. Tavakkoli-Moghaddam, “A hybrid algorithm based on particle swarm optimization and simulated annealing for a periodic job shop scheduling prob- lem,” The International Journal of Advanced Manufacturing Technology, vol. 54, no. 1–4, pp. 309–322, 2011.
15. J. Gao, L. Sun, and M. Gen, “A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems,” Computers and Operations Research, vol. 35, no. 9, pp. 2892–2907, 2008.
16. N. Al-Hinai and T. Y. Elmekkawy, “Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm,” International Journal of Production Economics, vol. 132, no. 2, pp. 279–291, 2011.
17. G. Zhang, X. Shao, P. Li, and L. Gao, “An effective hybrid particle swarm optimization algorithm for multi-objective flexible job- shop scheduling problem,” Computers& Industrial Engineering, vol. 56, no. 4, pp. 1309–1318, 2009.
18. W. Teekeng, A. Thammano, P. Unkaw, and J. Kiatwuthiamorn, “A new algorithm for flexible job-shop scheduling problem based on particle swarm optimization,” Artificial Life and Robotics, vol. 21, no. 1, pp. 1–6, 2016.
19. L. N. Xing, Y. W. Chen, P. Wang, Q. S. Zhao, and J. Xiong, “A knowledge-based ant colony optimization for flexible job shop scheduling problems ,”Applied Soft Computing, vol.10, no.3, pp. 888–896, 2010.
20. .-Q. Li, Q.-K. Pan, and M. F. Tasgetiren, “A discrete artificial bee colony algorithm for the multi-objective flexible job- shop scheduling problem with maintenance activities,” Applied Mathematical Modelling. Simulation and Computation for Engi- neering and Environmental Systems, vol. 38, no. 3, pp. 1111–1132, 2014.
21. P. Korytkowski, S. Rymaszewski, and T. Wi˚aniewski, “Ant colony optimization for job shop scheduling using multi- attribute dispatching rules,” The International Journal of Advanced Manufacturing Technology, vol. 67, no. 1–4, pp. 231– 241, 2013.
22. M.-S. Lu and R. Romanowski, “Multi-contextual ant colony optimization of intermediate dynamic job shop problems,” The International Journal of Advanced Manufacturing Technology, vol. 60, no. 5–8, pp. 667–681, 2012.
23. Adeyemo, I., Ojo, J., & Adegbola, O. Performance Evaluation of Three Evolutionary Algorithms for Selective Harmonic Elimination in Voltage Source Multilevel Inverter.
24. C.-J. Liao, Y.-L. Tsai, and C.-W. Chao, “An ant colony optimization algorithm for set up coordination in a two-stage production system,” Applied Soft Computing, vol. 11, no. 8, pp. 4521–4529, 2011.
25. Kamatchi, N., & Saravanan, D. A Hybrid Algorithm using Firefly and Cuckoo Search Algorithm for Flexible Open Shop Scheduling Problem.
26. W. Qin, J. Zhang, and D. Song, “An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time,” Journal of Intelligent Manufacturing, pp. 1–14, 2015.
27. R.-H. Huang, C.-L. Yang, and W.-C. Cheng, “Flexible job shop scheduling with due window—a two-pheromone ant colony approach,” International Journal of Production Economics, vol. 141, no. 2, pp. 685–697, 2013.
28. K.-L. HuangandC.-J. Liao,“Antcolony optimization combined with taboo search for the job shop scheduling problem,” Computers and Operations Research, vol. 35, no. 4, pp. 1030– 1046, 2008.
29. B. Zhao, J. Gao, K. Chen, and K. Guo, “Two-generation Pareto ant colony algorithm for multi-objective job shop scheduling problem with alternative process plans and unrelated parallel machines,” Journal of Intelligent Manufacturing, vol. 26, no. 3, pp. 1–16, 2015. Scientific Programming 11
30. J.-P. Arnaout, R. Musa, and G. Rabadi, “A two-stage ant colony optimization algorithm to minimize the makes panonun related parallel machines—Part II: enhancements and experimentations,”Journal of Intelligent Manufacturing, vol.25, no.1, pp.43– 53, 2014.

31. *Devi, K. D., Babu, K. S., & Reddy, K. H. Optimization of Cutting Force and Tool Temperature using Ann Based Multi Objective Genetic Algorithms In Turning Heat Treated Beryllium Copper Alloy.*

32. *C. W. Leung, T. N. Wong, K. L. Mak, and R. Y. K. Fung, "Integrated process planning and scheduling by an agent-based ant colony optimization, "Computers and Industrial Engineering, vol. 59, no. 1, pp. 166–180, 2010.*

33. *X. LiandL. Gao,"An effectivehy bridgenetic algorithm and tabu search for flexible job shop scheduling problem," International Journal of Production Economics, vol. 174, pp. 93–110, 2016.*

34. *L. Sun, L. Lin, Y. Wang, M. Gen, and H. Kawakami, "ABayesian optimization-based evolutionary algorithm for flexible job shop scheduling," Procedia Computer Science, vol. 61, pp. 521–526, 2015.*

35. *M. Mirabi, "Ant colony optimization technique for the sequence-dependent flow shop scheduling problem," The Inter- nationa l Journal of Advanced Manufacturing Technology, vol.55, no. 1–4, pp. 317–326, 2011.*

36. *C. Blum, "Ant colony optimization: introduction and recent trends," Physics of Life Reviews, vol. 2, no. 4, pp. 353–373, 2005.*

37. *Q. Ding, X. Hu, L. Sun, and Y. Wang, "An improved ant colony optimization and its application to vehicle routing problem with time windows," Neuro computing, vol. 98, pp. 101–107, 2012.*

38. *P. Surekha and S. Sumathi, "Solving fuzzy based job shop scheduling problems using GA and ACO," Journal of Emerging Trends in Computing and Information Sciences, vol. 1, no. 2, pp. 95–102, 2010.*

39. *T. St¨utzle and H. H. Hoos, "MAX-MIN ant system," Future Generation Computer Systems, vol. 16, no. 8, pp. 889–914, 2000.*

40. *I. Kacem, S. Hammadi, and P. Borne, "Approach by localization and multi objective evolutionary optimization for flexible job- shop scheduling problems," IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 32, no. 1, pp. 1–13, 2002.*

41. *M. Yazdani, M. Amiri, and M. Zandieh, "Flexible job-shop scheduling with parallel variable neighborhood search algorithm," Expert Systems with Applications, vol. 37, no. 1, pp. 678– 687, 2010.*

42. *L. Wang, S. Wang, Y. Xu, G. Zhou, and M. Liu, "Abi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem," Computers & Industrial Engineering, vol. 62, no. 4, pp. 917–926, 2012.*

43. *J.-Q. Li, Q.-K. Pan, and K.-Z. Gao, "Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems," The International Journal of Advanced Manufacturing Technology, vol. 55, no. 9, pp. 1159–1169, 2011.*

44. *T.-C. Chiang and H.-J. Lin, "A simple and effective evolutionary algorithm for multi objective flexible job shop scheduling," International Journal of Production Economics, vol.141, no.1, pp. 87–98, 2013.*

45. *N. B. Ho, J. C. Tay, and E. M.-K. Lai, "An effective architecture for learning and evolving flexible job-shop schedules, "European Journal of Operational Research, vol.179, no.2, pp.316–333,2007.*

46. *F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," Computers and Operations Research, vol. 35, no. 10, pp. 3202–3212, 2008.*